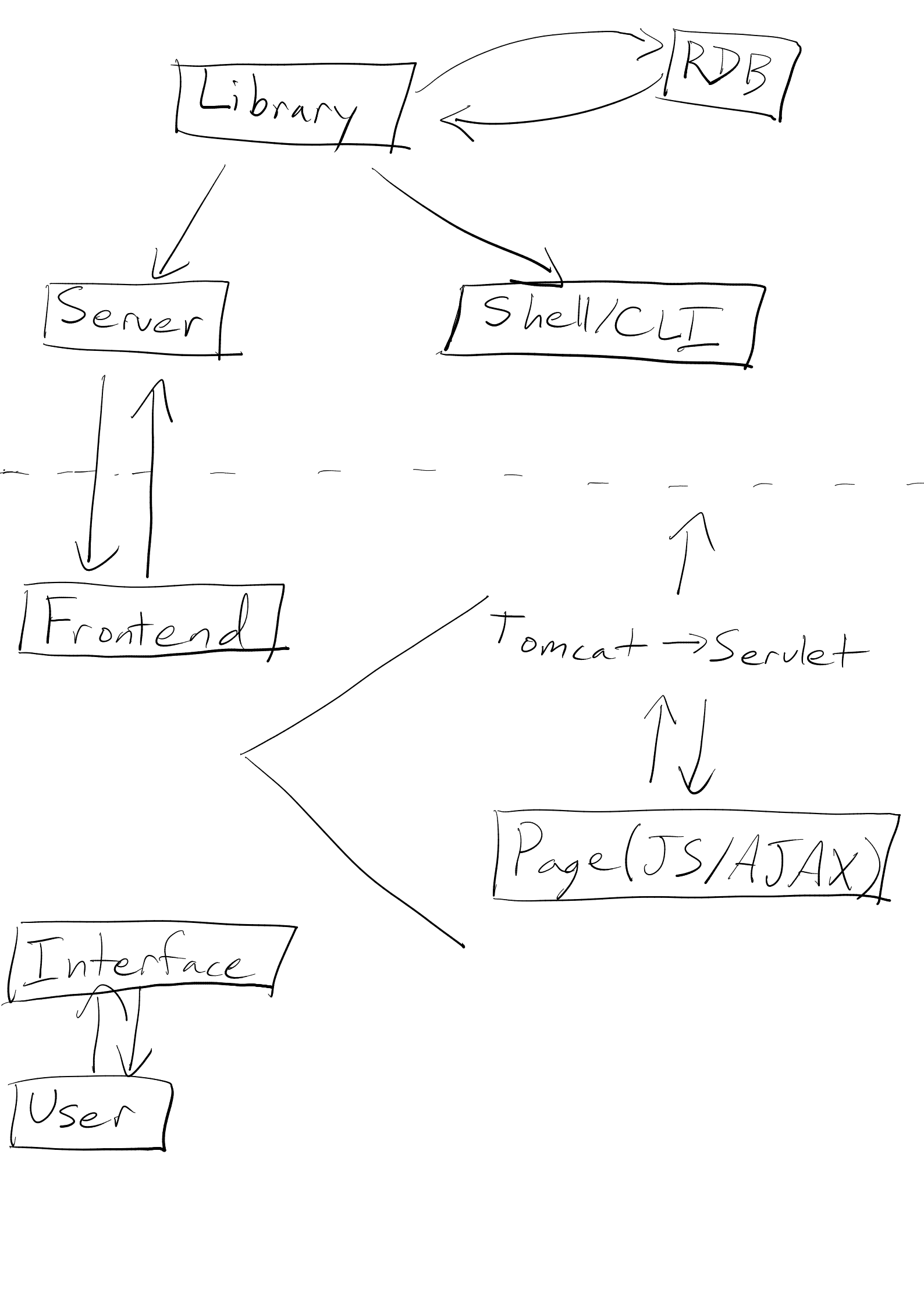

Library
RDB
Server
Shell/CLI
Frontend
Tomcat → Servlet
Page(JS/AJAX)
Interface
User

Algorithm

- Check - membership test
  - lookup
  - SELECT o trie

- Correction

List of Candidates ⇒ Edit distance

Filter

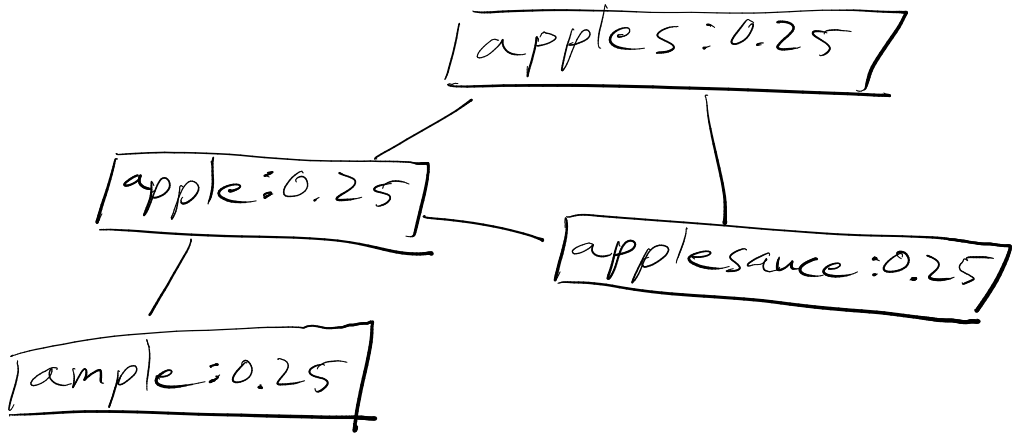
- First letter
- Length

Problems

- too many
- learning?

Alternative

- dist():
  - edit dist
  - first letter
  - length
  - frequency
  - ⋮



Graph

- Edit distance under threshold

Alg

- Find candidate/entry in graph
- Traverse, find lowest dist

- Find candidate
  - Filter
    - first letter
    - length
    - frequency?
    - random, dist under threshold

Traverse

- Dist of all neighbors
- Recurse on lowest dist

Edit Dist

- Dynamic Programming
- LRU Cached
- Insert
- Delete
- Replace (variable cost)
  - Generated with map (QWERTY) and Dikey's alg

Learning

- Add new words
- Modify frequencies

Data Structures

Trie?

NodeMap "a": Relation(Node next, extra chars)
"b":

Ex) apple, app

RDB (MySQL)

- Abstracts algorithms

words

id	word	length	frequency

graph

word1	word2

Performance

Time

- Add new word : $O(n)$ of $O(n^2)$
- Modify frequency : $O(1)/O(n)$
- Check/lookup : $O(\log n)/O(n)$
- Correct : ?? (fast)

Quality

??

- Time doesn't matter for end user
  - Server requests are threaded
  - Frontend atomizes words and handles request separately